**CENTRO UNIVERSITÁRIO BELAS ARTES DE SÃO PAULO**

**GRADUAÇÃO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - EAD**

**FELIPE SCHAITEL - 24112424**

**PROCESSO DE DESENVOLVIMENTO DE SISTEMAS - ATIVIDADE 01**

**FLORIANÓPOLIS, SC**

**2025**

**TRABALHO – N1**

QUESTÕES

1. Valor (0,50 ponto) – Explique a importância dos aspectos humanos na engenharia de software e como eles influenciam tanto o processo de desenvolvimento quanto o produto final de gestão.

A terminologia “Engenharia de *Software*” pode passar a sensação de que é um campo de conhecimento puramente tecnológico. Entretanto, os aspectos humanos são importantes e fazem parte tanto da engenharia quanto para o engenheiro de *software*.

Segundo Roger S. Pressman no Livro “Engenharia de *Software*”

*Software* é desenvolvido por pessoas, usado por pessoas e dá suporte à interação entre pessoas. Assim, características, comportamento e cooperação humanos são fundamentais no desenvolvimento prático de *software.*

(PRESSMAN, p. 74)

Existem diversos campos de conhecimento que dedicam suas pesquisas e estudos sobre como as pessoas interagem, reagem e experienciam as aplicações, sistemas e *softwares*. Pois essas soluções tecnológicas são para pessoas utilizarem e por isso, uma boa interface que segue padrões de usabilidade, por exemplo, impactam diretamente no sucesso de um *software.* Então é muito importante considerar os aspectos humanos tanto dos usuários quanto da equipe e até mesmo o pessoal para o processo de análise dos requisitos, planejamento da arquitetura, *design*, desenvolvimento, teste e implementação de *software*.

2. Valor (0,50 ponto) – Dentre as sete características pessoais identificadas por Erasmus que tornam um engenheiro de software competente, escolha 3 que você considere mais importantes e explique o motivo dessa escolha.

Roger S. Pressman no Livro “Engenharia de *Software*” afirma que existem aspectos humanos importantes na engenharia de *Software*. Esses aspectos são importantes para tornar o engenheiro de *Software* competente. Estes sete aspectos, características pessoais, foram identificados por *Erasmus* e estão presentes em engenheiros de *Software* competentes:

1. Responsabilidade Individual;
2. Consciência aguçada das necessidades;
3. Extremamente honesto;
4. Resiliência sob pressão;
5. Elevado senso de lealdade;
6. Atenção aos detalhes;
7. Pragmático;

Considero todas essas características importantes e fundamentais mas, as três características que eu considero mais importantes entre as sete são: Responsabilidade individual, consciência aguçada e atenção aos detalhes.

Pois é importante ter o senso de responsabilidade para cumprir e fazer o que foi prometido, assim não sobrecarrega e/ou atrasa os colegas. Ter a consciência aguçada é importante para entender as necessidades de todos os envolvidos no projeto, desde os *stakeholders*, usuários, equipe etc.

Por último, atenção aos detalhes pois percebo que essa característica impacta diretamente na qualidade do processo de desenvolvimento do *software* e do produto, ajuda a evitar erros. A capacidade de observar detalhes podem fazer uma diferença grande no sucesso e desempenho do *software*, além de fazer a entrega ter mais confiabilidade e ser de alto nível.

3. Valor (0,50 ponto) – Discuta os atributos normalmente encontrados em equipes de software eficazes.

No material didático da disciplina ensina que equipes eficazes precisam ter consistência, que consiste em todos estarem com o mesmo objetivo em comum, propósito, envolvimento, confiança e a busca por melhoria constante do trabalho. Essas são as características, estruturas e dinâmicas de equipes eficazes. Equipes que estão em sintonia com um mesmo alinhamento e sintonia em comum geram engajamento, motivação, fomenta a inovação, criatividade além de reduzir conflitos impactando na sensação de bem-estar dos membros da equipe. Esse fenômeno da consistência de equipe impacta no sucesso, manutenção e durabilidade do *software*.

Há muitos exemplos de equipes consistentes que desenvolvem *softwares* como a equipe de *Linux Kernel* em que a comunidade mundial de desenvolvedores colaboram e seguem o mesmo propósito: manter e melhorar o [[1]](#footnote-0)*kernel* da *Linux.* Há também a equipe da *Mozilla (Firefox),* time do *Google Chrome* etc.

Mas o mais interessante sobre equipes consistentes é que esse é um fenômeno que perpassa o campo da engenharia de *Software*. Por exemplo, a *Pixar Animation Studios* tem o propósito em comum de criar animações com a maior qualidade e com novas tecnologias.

Um exemplo brasileiro de equipe consistente é a equipe de dubladores do *anime Yu-yu-Hakusho*, é possível de notar nas entrevistas que os membros estavam em sintonia, felizes, se divertindo, com o mesmo propósito, objetivo, imersão e comprometimento em comum e até hoje essa é considerada uma das mais memoráveis e melhores dublagens de desenho brasileira. Mais de 20 anos depois do lançamento, os internautas ainda compartilham conteúdo e *meme* referente a essa dublagem. Hoje em dia a equipe está reunida junto com Wendel Bezerra (dublador do *Goku* de *Dragon Ball*) trabalhando na dublagem de *One Piece.* Em menos de 2 anos a equipe dublou aproximadamente 750 episódios e estão produzindo ainda mais.

Esses exemplos demonstram o como equipes eficazes são produtivas, geram artefatos consistentes e proporcionam um ambiente de trabalho harmonioso.

Equipes consistentes têm muito potencial e criam produtos com muita consistência, qualidade, desempenho, capricho, eficiência e que podem durar por anos.

4. Valor (0,50 ponto) – Analise o impacto das mídias sociais no trabalho de engenharia de software, incluindo benefícios e possíveis preocupações relacionadas à privacidade e segurança.

Quando um Software cresce é normal que a equipe de desenvolvimento aumente de tamanho junto. Por isso, as mídias sociais têm um papel importante para facilitar, gerenciar e apoiar a comunicação. Segundo Pressman “*o uso de mídias sociais (...) o gerenciamento de versões que pode proporcionar melhorias significativas na maneira como desenvolvedores e clientes se comunicam*” página 603. Redes sociais como *Linkedin* é um exemplo de mídia social que auxilia engenheiros, DevOps etc de se conhecerem, comunicarem, criarem conexões e equipes produtivas.

É fundamental a ética, privacidade e segurança ao se usar as mídias sociais para não expor dados e informações sigilosas/ privativas.

5. Valor (0,50 ponto) – Explique 2 dos 5 fatores responsáveis pela toxicidade de uma equipe.

Segundo o material didático da disciplina, os cinco fatores responsáveis pela toxicidade de uma equipe são:

1. Atmosfera de trabalho frenética;
2. Alto grau de frustração e atrito na equipe;
3. Processo de software fragmentado mal coordenado;
4. Contínua exposição das falhas;
5. Pouca definição dos papeis e funções de cada membro da equipe.

Uma forma possível de resolver a toxicidade da atmosfera de trabalho frenética é dar acesso para a equipe de todas as informações que são necessárias para completar a tarefa. Depois que os objetivos e metas forem definidos é importante que não mudam toda hora, apenas se for muito necessário.

A toxicidade de alto grau de frustração e atrito na equipe pode ser solucionado e evitado através da escuta e comunicação. Os funcionários precisam ver que sua experiência e conselhos são levados em consideração nas decisões. Por isso, sempre que possível oferecer a responsabilidade para tomadas de decisões.

6. Valor (0,50 ponto) – Discuta como a diversidade em uma equipe de software pode melhorar a eficácia da equipe e a qualidade do software produzido.

A diversidade é importante em uma equipe de *software* porque combinam variedades de diferentes qualidades, conhecimentos e experiências. Segundo Pressman na página 76 *“As equipes de software mais eficazes são diversificadas*” .

7. Valor (0,50 ponto) – Explique como uma comunicação eficaz dentro de uma equipe de software contribui para o sucesso do projeto de desenvolvimento.

De acordo com o material didático a comunicação eficaz junto de uma liderança ética de equipes bem estruturadas são circunstâncias para o sucesso. A comunicação e a colaboração são as primeiras etapas a serem feitas no início de um projeto. Para compreender as necessidades e objetivos dos envolvidos, a análise e documentação de requisitos, que irá definir os recursos, funcionalidades etc do *software.* AComunicação eficaz resulta em requisitos bem estruturados que junto a arquitetura de software são o “esqueleto” do sistema. O planejamento, o *layout*, a codificação etc irá construir as funcionalidades do que foi projetado a partir dos requisitos.

Então uma má comunicação pode impactar negativamente as funcionalidades, o desempenho, o banco de dados etc de umsistema. Portanto, a equipe se comunicar entre si, também com *stakeholders* e usuários é fundamental.

8. Valor (0,50 ponto) – Explique a importância de ser 'extremamente honesto' no ambiente de engenharia de software e como isso pode ser feito sem ser percebido como insultante ou agressivo.

Ser extremamente honesto é uma característica de Engenheiros de Software competentes pois aponta os defeitos e o que pode se melhorar quando se percebe uma falha. De maneira honesta, realista, sincera e construtiva. A maneira como é dito importa muito para criar um ambiente de trabalho harmônico e não gerar a toxicidade da exposição constante de falhas. Apontar acertos e erros requer cordialidade, cuidar com o tom da voz e gentileza são características necessárias para elaborar uma crítica construtiva que visa a melhoria.

9. Referências de leituras

1. Consulte os materiais adicionados em cada aula.

2. PRESSMAN, Roger S.; MAXIM, Bruce. Engenharia de software. 9a edição. São Paulo: AMGH, 2021 3. SOMMERVILLE, I. Engenharia de software. 10. ed. São Paulo: Pearson Prentice Hall, 2019.

4. SOFTWARE ENGINEERING INSTITUTE (SEI). CMMI for Development, Version 1.3 CMMIDEV, V1.3 – CMMI Product Team – Improving processes for developing better products and services – Technical Report. Pittsburgh: Carnegie Mellon University, 2010.

1. kernel: é o principal componente de um sistema operacional Linux. [↑](#footnote-ref-0)